

REMARKS

Claims 1-30 are pending in this application. In an office action dated May 12, 2003, the Examiner rejected Claims 1-20. Claims 5-8, 10-13, and 18-20 have been amended. Claims 21-30 are new

Amendments and New Claims

Claims 5, 11, and 18 have each been amended to more clearly indicate that the third party legacy data list comprise one or more list of names and telephone numbers stored in a format that is not otherwise compatible with the application program.

Claims 6, 12, and 19 have been amended to more clearly indicate that the third party legacy data list comprise one or more list of names and e-mail addresses stored in a format that is not otherwise compatible with the application program. Claims 7, 13, and 20 have been amended to more clearly indicate that the third party legacy data list comprise contact information and appointment information stored in a format that is not otherwise compatible with the application program. These amendments all find support in the Specification as filed on page 1, lines 8-12 and page 4, lines 6-14.

Claim 8 has been amended to correct a misspelling. Claim 10 has been amended by deleting an unneeded prepositional phrase. New Claims 21-27 find support in the Specification starting on page 7, line 4 and continuing through page 9, line 11. Claims 28-30 find support in the section noted just above as well as on page 5, line 17 through page 6, line 19.

Abstract

The Examiner objected to the abstract of the disclosure because it contained more than 150 words. The Abstract has been amended addressing the Examiner's concerns.

Specification

The Examiner objected to the specification noting a misspelling and a grammatical error. The specification has been amended addressing the Examiner's concerns.

Claim Objection

The Examiner objected to Claim 8 noting a misspelling. Claim 8 has been amended addressing the Examiner's concerns.

Rejection under 35 U.S.C. § 112

The Examiner rejected Claim 10 arguing that a term lacked proper antecedent basis. Claim 10 has been amended addressing the Examiner's concern.

Rejections Under 35 U.S.C. § 102

The Examiner rejected Claims 1-4, 8-10, and 14-17 under Section 102(b) citing Hendrickson (USPN 5,933,646). A claim is anticipated if, and only if, each and every limitation set forth in the claim can be found expressly or inherently in a single piece of prior art. *Verdegaal Bros. V. Union Oil Co. of California* 814 F.2d 628, 631 (Fed. Cir. 1987).

CLAIMS 1-4: This first group of claims are directed to a method for providing an access to one or more third party legacy data list to a user of an application program. Claim 1 is independent. Claims 2-4 depend from Claim 1. Claim 1 requires the following limitations:

1. querying an operating system, by said application program upon start of said application program, whether one or more plug-in module is registered in a registry of an operating system, said one or more plug-in modules being capable of interfacing with corresponding respective ones of said one or more third party legacy data list;
2. receiving, from said one or more plug-in modules found in said registry, identifications of ones of said one or more third party legacy data list corresponding to said found one or more plug-in modules; and
3. providing a list of said identifications to said user through a user interface of said application program.

Hendrickson discloses a software manager that enables a computer user to administer software elements in a computer system. Hendrickson, Abstract. Addressing the first limitation above, the Examiner cited Hendrickson, col. 3, lines 4-

16, and col. 6, line 47 through col. 7, line 17. The cited sections do discuss the use of plug-ins. However, the plug-ins discussed are designed to provide an interface between the software-manager server. Hendrickson, col. 3, lines 6-8. More specifically, the plug-ins the software components to request and receive notification of certain changes in the information stored in a configuration database. Hendrickson, col. 3, lines 8-11.

Each of the plug-ins of Hendrickson provide an interface between a software component and software manager server. See Hendrickson, col. 7, lines 5-29. More specifically, each of Hendrickson's plug-ins is associated with a given software component. Hendrickson, col. 7, lines 8-9. The plug-in for a given software component provides information to the software manager server that describes a hierarchical between that software component and other software components as well as a listing of system file content that affects that software component. Hendrickson, col. 7, lines 8-14. When the software manager server detects a change in a file system that will affect a given software component, it notifies the plug-in for that software component which in turn notifies that software component. Hendrickson, col. 7, lines 24-29.

Nothing in the cited sections teaches querying an operating system registry for a plug-in capable of interfacing with a third party legacy data list as required by the first limitation.

Addressing the second limitation above, the Examiner cited Hendrickson, col. 7, lines 8-17. As already noted above, the cited section describes that Hendrickson's plug-ins are each associated with a software component. Further, the plug-in for a given software component is capable of proving a software manager server with information describing the hierarchical relationship between that software component and other software components. The plug-in for a given software component is also responsible for providing the software server manager with a listing of system file content that affects that software component.

Nothing in the cited section teaches receiving, from a plug-in module, an identification of a third party legacy data list corresponding to that plug-in. Instead, Hendrickson's plug-ins supply information regarding software components.

Addressing the third limitation above, the Examiner cited Hendrickson, col. 2, lines 40-52. The cited section describes a user interface that displays the status of software

components and any interdependencies between the software components. Nothing in the cited section teaches providing a list of identified third party legacy data lists.

For these reasons, Claim 1 is felt to distinguish over Hendrickson. Claims 2-4 each depend from Claim 1 and include all of the limitations of that base claim. As such, Claims 2-4 are felt to distinguish over Hendrickson.

CLAIMS 8-10: This second group of claims are directed to a system version of the method(s) of claims 1-4. Claim 8 is independent. Claims 9 and 10 depend from Claim 8. Claim 8 requires the following limitations:

1. one or more plug-in module, each of which being:
 - a. capable of interfacing with an associated one of said one or more third party data list,
 - b. registered in a registry of an operating system of said computer system; and
2. an application program having a user interface, said application program:
 - a. upon starting being in communication with said operating system to query said registry to determine registered ones of said one or more plug-in module,
 - b. configured to query each of said registered ones of said one or more plug-in modules for names of said one or more third party legacy data list,
 - c. configured to provide a list of said names of said one or more third party legacy data list to said user through said user interface.

The Examiner asserts that, for the same reasons Claim 1 is anticipated by Hendrickson, so is Claim 8. It is presumed then that the Examiner bases the rejection of Claim 8 on Hendrickson, col. 2, lines 40-52., col. 3, lines 4-16, and col. 6, line 47 through col. 7, line 17. As pointed out above, nothing in these cited sections teaches a plug-in capable of interfacing with a third party legacy data list as required by the first limitation of Claim 8. Moreover, nothing in the cited sections teaches an application program capable of querying plug-in modules to identify third party legacy data lists.

For these reasons, Claim 8 is felt to distinguish over Hendrickson. Claims 9 and 10 each depend from Claim 8 and include all of the limitations of that base claim. As such, Claims 9 and 10 are felt to distinguish over Hendrickson.

CLAIMS 14-17: This third group of claims are directed to computer readable medium with instructions for performing the method(s) of claims 1-4. Claim 14 is independent. Claims 15-17 depend from Claim 8. Claim 15 requires a computer readable medium with instructions for performing the following limitations:

1. querying an operating system, by said application program upon start of said application program, whether one or more plug-in module is registered in a registry of an operating system, said one or more plug-in modules being capable of interfacing with corresponding respective ones of said one or more third party legacy data list;
2. receiving, from said one or more plug-in modules found in said registry, identifications of ones of said one or more third party legacy data list corresponding to said found one or more plug-in modules; and
3. providing a list of said identifications to said user through a user interface of said application program.

The Examiner asserts that, for the same reasons Claim 1 is anticipated by Hendrickson, so is Claim 14. It is presumed then that the Examiner bases the rejection of Claim 8 on Hendrickson, col. 2, lines 40-52., col. 3, lines 4-16, and col. 6, line 47 through col. 7, line 17. As pointed out above, nothing in these cited sections teaches a plug-in capable of interfacing with a third party legacy data list as required by the first limitation of Claim 8. Moreover, nothing in the cited sections teaches receiving from plug-in modules identifications of third party legacy data lists.

For these reasons, Claim 14 is felt to distinguish over Hendrickson. Claims 15-17 each depend from Claim 8 and include all of the limitations of that base claim. As such, Claims 15-17 are felt to distinguish over Hendrickson.

Rejections Under 35 U.S.C. § 103

The Examiner rejected Claims 5-7, 11-13, and 18-20 under Section 103(a). To establish a prima facie case of obviousness under Section 103, the Examiner

must show some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings; that there is a reasonable expectation of success; and that the prior art reference (or references when combined) teach or suggest all the claim limitations. MPEP § 2142 (*citing In re Vaeck*, 947 F.2d 488, (Fed. Cir. 1991)).

CLAIMS 5, 11, AND 18: The Examiner rejected these claims as being unpatentable over Hendrickson in view of Tsukamoto (USPN 5,857,073). Claim 5 depends from Claim 4 which depends from Claim 1. Claim 11, depends ultimately from Claim 8, and Claim 18 depends ultimately from Claim 15. For the same reasons Claims 1, 8 and 15 distinguish over Hendrickson, so do Claims 5, 11, and 18.

Nonetheless, Claim 5, requires the following additional limitations not taught by Hendrickson or Tsukamoto:

1. the application program comprises a facsimile software; and
2. one or more third party legacy data list comprise one or more list of names and telephone numbers stored in a format that is not otherwise compatible with the application program using a legacy facsimile software.

Claim 11 is a system version of the method of Claim 5. Claim 18 is directed to a computer medium having instructions for performing the method of Claim 5.

Rejecting Claims 5, 11, and 18, the Examiner admits that Hendrickson does not teach the above limitations. Instead, with reference to the first limitation above, the Examiner cites Tsukamoto, col. 4, lines 10-18, col. 5, lines 16-65, col. 11, lines 14-16, and Fig. 5. The cited sections discuss a network made up client terminals, a facsimile server, a file server, and a number of facsimile devices. Tsukamoto, col. 5, lines 16-26. A facsimile device includes a mailbox group stored in its own RAM, and the mail box group contains a number of mail boxes. Tsukamoto, col. 5, lines 17-35. Each mail box contains a mail box number, a user name, an identification ID, and addresses of received facsimile messages. Tsukamoto, col. 5, lines 36-40.

Upon receipt of a facsimile message for a particular user, a facsimile device writes an address in the reception message area of the user's mail box and stores

the facsimile message in its memory at that address. Tsukamoto, col. 5, lines 41-46. 16-26 The facsimile server can then supply a user name and an identification, and the facsimile device will transfer the facsimile message to the facsimile server. Tsukamoto, col. 5, lines 46-49.

Fig. 5 illustrates a flow of changed registration data specific to a facsimile device. Tsukamoto, col. lines 54-55. Fig. 5 illustrates "setting data" stored in the RAM of a facsimile device and consisting of a name, a telephone number, and an auto reception indicator. Tsukamoto, col. 9, lines 63-65. Also illustrated is a general data group stored on the facsimile server. The setting data is been changed and differs from the general data. The general data is updated (presumably by some nondescript software operating on the facsimile server) to reflect changes made to the setting data. Tsukamoto, col. 10, lines 52-56.

While it is not clear, it is assumed that the Examiner is equating nondescript software running on a facsimile server with the facsimile software of Claim 5. It is also assumed that the Examiner is equating the general data with the list of names and telephone numbers making up a legacy data list of Claim 5. Claim 5 has been amended to clearly indicate that the legacy facsimile software (used to store the names and telephone numbers of a legacy data list) is different from the facsimile software that is the application program. The nondescript software running on Tsukamoto's facsimile server is more akin to the legacy facsimile software of Claim 5.

Neither Hendrickson or Tsukamoto, alone or in combination, teach an application program that is a facsimile software and a legacy data list that is a list of names and telephone numbers stored by a legacy facsimile software that is different than the facsimile software that is the application program. For this reason, Claims 5, 11, and 18 are felt to distinguish over the cited art.

CLAIMS 6, 12, AND 19: The Examiner rejected these claims as being unpatentable over Hendrickson in view of Sampath (USPN 6,266,744). Claim 6 depends from Claim 4 which depends from Claim 1. Claim 12, depends ultimately from Claim 8, and Claim 19 depends ultimately from Claim 15. For the same reasons Claims 1, 8 and 15 distinguish over Hendrickson, so do Claims 6, 12, and 19.

Nonetheless, Claim 6, requires the following additional limitations not taught by Hendrickson or Sampath

1. the application program comprises an e-mail software; and
2. one or more third party legacy data list comprises one or more list of names and e-mail addresses stored in a format that is not otherwise compatible with the application program using a legacy e-mail software.

Claim 12 is a system version of the method of Claim 6. Claim 19 is directed to a computer medium having instructions for performing the method of Claim 6.

Rejecting Claim 6, the Examiner admits that Hendrickson does not teach the above limitations. Instead, with reference to the first limitation above, the Examiner cites Sampath, col. 4, lines 33-43. The cited section discusses a server computer that is programmed to generate invoices that can be sent, for example, via electronic mail. The cited section describes invoicing software not e-mail software as required by Claim 6. With reference to the second limitation, the Examiner cites Sampath, col. 6, lines 15-18. This section describes a billing database (connected to the server computer) that can contain an e-mail address. While not clear, it is presumed that the invoicing software on the server computer is responsible for storing the e-mail address in the billing database.

Neither Hendrickson or Sampath, alone or in combination, teach an application program that is an e-mail software and a legacy data list that is a list of e-mail addresses stored by a legacy facsimile software that is different than the facsimile software that is the application program. For this reason, Claims 6, 12, and 19 are felt to distinguish over the cited art.

CLAIMS 7, 13, AND 20: The Examiner rejected these claims as being unpatentable over Hendrickson in view of Gottsman (USPN 6,134, 548). Claim 7 depends from Claim 4 which depends from Claim 1. Claim 13, depends ultimately from Claim 8, and Claim 20 depends ultimately from Claim 15. For the same reasons Claims 1, 8 and 15 distinguish over Hendrickson, so do Claims 7, 13, and 20.

Nonetheless, Claim 7, requires the following additional limitations not taught by Hendrickson or Gottsman:

1. the application program comprises a personal organizer software; and
2. one or more third party legacy data list comprise one or more contact information and appointment information stored in a format that is not otherwise compatible with the application program using a legacy personal organizer software.

Claim 13 is a system version of the method of Claim 7. Claim 20 is directed to a computer medium having instructions for performing the method of Claim 7.

Rejecting Claim 7, the Examiner admits that Hendrickson does not teach the above limitations. Instead, with reference to the first limitation above, the Examiner cites Gottsman, col. 37, lines 9-12. The cited section reads as follows:

A Personal Digital Assistant (PDA) with Internet access can synchronize the person's calendar, email, contact list, task list and notes on the PDA with the version stored in the Internet site. This enables the person to only have to maintain one version of this data in order to have it available whenever it is needed and in whatever formats it is needed.

Neither Hendrickson or Gottsman, alone or in combination, teach an application program that is a personal organizer software and a legacy data list that is contact information and appointment information stored using a legacy personal organizer software that is different than the facsimile software that is the application program. For this reason, Claims 7, 13, and 20 are felt to distinguish over the cited art.

Conclusion

In view of the foregoing amendments and remarks, Applicant respectfully submits that Claims 1-30 are all in condition for allowance. Consequently, early and favorable action allowing these claims and passing the application to issue is earnestly solicited. The foregoing is believed to be a complete response to the outstanding Office Action.

AUG-07-2003(THU) 13:32

Ormiston & McKinney

(FAX)208 433 9295

P. 024/026

Respectfully submitted,
David Bohan

By


Jack H. McKinney
Reg. No. 45,685

August 7, 2003

CHANGES TO THE SPECIFICATION

OFFICIAL

The paragraph spanning lines 6 through 16 of page 2:

In accordance with another aspect of the present invention, a system for providing an access to one or more third party legacy data list to a user of a computer system comprises one or more plug-in module, each of which being capable of interfacing with an associated one of the one or more third party data list, each of the one or more plug-in modules being registered in a registry of an operating system of the computer system, and an application program having a user interface, the application program upon starting being in communication with the operating system to query the registry to determine registered ones of the one or more plug-in module, the application program further configured to query each of the registered ones of the one or more plug-in modules for names of the one or more third party legacy data list, the application program being configured to provide a list of the names of the one or more third party legacy data list test to the user through the user interface.

The paragraph spanning lines 15 through 28 of page 4:

In one embodiment, the plug-in modules are compiled as and a dynamic linked library (DLL) and registered in the operating system registry based on an object oriented specification, e.g., the component object model (COM) specification provided by the Microsoft Corporation of Redmond, Washington, USA. The plug-in modules are listed under a COM category in the registry of the operating system, and are discovered by the application program at start up. Additional plug-in modules may be added at any time simply by registering the additional plug-in modules. The next time the application program is started, the additional plug-in modules are automatically discovered. Using the discovered plug-in modules, the application program allows the user to access and/or edit the third party legacy data lists through the user interface of the application program. In an alternative embodiment, upon an indication, by the user of the application program, a desire to edit data of a particular third party legacy data list, the corresponding plug-in module

causes the editor user interface screen of the legacy data list to be displayed to the user. The user is allowed to edit the data list using the user interface of the legacy data list.

The Abstract:

~~A method and system of a computer application program allows a user of the application program to utilize third party legacy data lists, e.g., contact information lists, telephone and facsimile numbers, scheduling and/or appointment information or the like, through a common interface, without requiring the user to manually type in the data into a new data list or to maintain multiple data lists. For each of the third party legacy data list installed on the computer, one of a plurality of plug-in modules also installed on the computer acts as an interface between the application program and the legacy data list. Each of the plug-in module is an object accessible by the application program at run time. In one embodiment, the plug-in modules are compiled and a dynamic linked library (DLL) and registered in the operating system registry based on an object-oriented specification, e.g., the component object model (COM) specification provided by the Microsoft Corporation of Redmond, Washington, USA. The plug-in modules are listed under a COM category in the registry of the operating system, and are discovered by the application program at start up. Additional plug-in modules may be added at any time simply by registering the additional plug-in modules. The next time the application program is started, the additional plug-in modules are automatically discovered. Using the discovered plug-in modules, the application program allows the user to access and/or edit the third party legacy data lists through the user interface of the application program, or through an edit user interface of the third party legacy data list. An application program is provided access to a third party legacy data list not supported by the application program. A method embodiment includes supplying a plug-in module capable of providing an interface between the application program and the third party legacy data list. The application program sends a function call to the plug-in module. The plug-in module, in response to the function call, accesses the third party legacy data list and provides the application program information relating to the third party legacy data list.~~